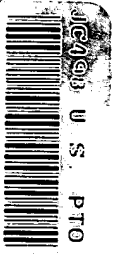


12/04/98



ATTORNEY'S DOCKET NO. P3557
PATENTS

**UNITED STATES PATENT APPLICATION
OF**

GERALD J. HANAM

FOR

**SYSTEM AND METHOD FOR FACILITATING DISCOVERY OF NETWORK ADDRESSES AND
SELECTED CHARACTERISTICS OF COMPUTER SYSTEMS AND THE LIKE WHICH ARE
CONNECTED IN DIGITAL DATA NETWORKS**

Certificate of Express Mailing

Express Mail Mailing Label No. EE 819 407 929 US

Date of Deposit December 4, 1998

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office To Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231.

By Richard A. Jordan

Richard A. Jordan

-1-

FIELD OF THE INVENTION

The invention relates generally to the field of digital data networks, and more particularly to systems and methods for facilitating discovery of network addresses and selected characteristics of computer systems and the like which are connected to such networks.

BACKGROUND OF THE INVENTION

In modern "enterprise" digital data processing systems, that is, computer systems for use in an office environment in a company, a number of personal computers, workstations, and other devices such as mass storage subsystems, network printers and interfaces to the public telephony system, are typically interconnected in a computer network. The personal computers and workstations (generally, "computers") are used by individual users to perform processing in connection with data and programs that may be stored in the network mass storage subsystems. In such an arrangement, the computers, operating as clients, access the data and programs from the network mass storage subsystems for processing. In addition, the computers will enable processed data to be uploaded to the network mass storage subsystems for storage, to a network printer for printing, to the telephony interface for transmission over the public telephony system, or the like. In such an arrangement, the network mass storage subsystems, network printers and telephony interface operate as servers, since they are available to service requests from all of the clients in the network. By organizing the network in such a manner, the servers are readily available for use by all of the computers the network. Such a network may be spread over a fairly wide area, with the computers being interconnected by communication links such as electrical wires or optic fibers.

A problem arises when an operator at one computer wishes to enable the computer to determine network addresses that are used by other computers and other resources (generally, "computers") for communications over the network, and selected characteristics of those computers. For example, an operator of one computer connected to the network may wish to use programs or

-2-

1 data which are on other computers (including both client computers and server computers) which
2 are connected to the network, in which case he or she may need to initially obtain the network
3 addresses which identify the particular computers which are connected to the network. After
4 obtaining the identifications, the operator may retrieve information from or transfer information to
5 their respective information storage systems. In addition, after obtaining the identifications the
6 operator which may enable ones of the other computers to perform, for example, processing
7 operations in connection with remote procedure calls (RPC's) transmitted thereto over the network.

8 Several general methodologies have been used to enable an operator to determine the various
9 computers which are available over the network. In one methodology, each computer can be
10 provided with one or more lists of other computers and other resources which are available to it over
11 the network. In such a case, the list would need to be updated by, for example, a system
12 administrator as each computer or other resource is added to or removed from the network. There
13 are several problems with this methodology. First, maintaining all of the lists in an updated
14 condition can be a difficult task, particularly as the size of the network increases and computers and
15 other resources are added or removed in remote portions of the network. In addition, such lists
16 would normally only identify the particular computers and resources which are connected to the
17 network, and would not necessarily indicate whether they are available for use at any particular point
18 in time; thus, for example, if a particular computer or other resource is switched off, it would still
19 be identified in a list but would not actually then be available.

20 In another methodology, each computer periodically broadcasts "advertising" messages
21 containing their respective identifications over the network, perhaps along with other information
22 such as various types of resources that they may provide which may be accessed over the network.
23 The various computers can receive the broadcast messages and from the information contained
24 therein can identify the computers and other resources which broadcast the respective messages. The
25 computers can buffer the information that they receive in the respective broadcast messages and
26 provide it to the operator when requested. Since new computers may be added to the network, the
27 computers which broadcast the messages will need to broadcast them periodically over the network.

1 While this methodology allows each computer to identify other computers which are available over
2 the network, without requiring assistance from a system administrator or other operator, it does
3 require the computers and other resources to broadcast a number of messages. A number of the
4 broadcast messages may be unnecessary, either because the computers and other resources which
5 are connected to and available over the network has not changed, or because operators have not
6 requested information as to the computers and resources which are available over the network. This
7 problem is exacerbated as the number of computers which are connected to the network increases,
8 since the number of advertising messages would increase correspondingly and reduce the network
9 bandwidth which is available for transfer of other information over the network. Additionally for
10 networks which are interconnected by devices such as routers, the broadcast advertising messages
11 may not be transferred throughout the entire network, since typically routers are designed so as not
12 to forward broadcast messages.

13 A further methodology has been developed which alleviates these problems to some extent.
14 In this further methodology, when an operator at a computer wishes to know what other computers
15 and other resources are available over the network, his or her computer will broadcast an inquiry
16 message over the network. Other computers connected to the network are to respond to the inquiry
17 message with the required information, including, for example, their respective identifiers, within
18 a particular time-out period. The computer that had broadcast the inquiry message will receive the
19 information can provide the information which it receives during the time-out period to the operator.
20 While this methodology alleviates the transmission of unnecessary messages which will be broadcast
21 in the previously-described methodology, and it ensures that the information provided to the operator
22 is current, it does have several problems. First, the requirement that a computer or other resource
23 respond within the time-out period in order for its information to be included in the information
24 displayed to the operator, results in information not being included if it is not received within the
25 required time-out period. Thus, if a computer or other resource is busy and not able to respond to
26 the broadcast message within the time-out period, its information will not be provided to the operator
27 even though it may be available for use. Furthermore, if there are a large number of computers

1 connected to the network which will respond to the broadcast, the computer that broadcast the
2 message may be inundated with response messages, which can cause congestion and lost messages.

3 To alleviate the congestion problem, instead of a broadcast message, the computer that is
4 determining the network addresses being used by other computers in the network can transmit
5 individual address inquiry message packets using each of the possible network addresses, or a
6 predetermined subset thereof. If another computer receives an address inquiry message packet, it
7 will generate a response message packet for transmission to the computer that transmitted the
8 address inquiry message packet, which response message packet includes its network address. The
9 computer could transmit a single address inquiry message packet, wait for a predetermined time
10 interval, during which it may get a response, and then transmit another address inquiry message
11 packet, and repeat these operations for each network address to be tried. However, operating in this
12 manner can require an unduly long time to complete operations in connection with all of the network
13 addresses to be tried, particularly if the time interval is relatively long, which may be the case if the
14 network is large and worst-case propagation delays may be commensurately large. To accommodate
15 that, the computer can transmit a plurality of address inquiry message packets, and wait for a
16 predetermined time period before sending out additional address inquiry message packets. This can
17 still require an unduly long period of time, particularly if the network is large and the time interval
18 is relatively long.

19 SUMMARY OF THE INVENTION

20 The invention provides a new and improved system and method for facilitating efficient
21 determination by a computer connected in a computer network of network addresses used by other
22 computers and other devices connected in the network, and of determining predetermined
23 characteristics of those other computers and other devices.

1 In brief summary, a system for use by a computer to determine which network addresses are
2 being used by other computers or other devices in a network includes an address inquiry message
3 packet generator module configured to enable the computer to transmit address inquiry message
4 packets over a network in one or more iterations. Each address inquiry message packet includes a
5 network address. The address inquiry message packet generator module is configured to enable the
6 computer to transmit address inquiry message packets for a selected number of network addresses
7 during each iteration. An iteration control module is configured to control the timing of successive
8 iterations of transmission of address inquiry message packets by the computer in relation to reception
9 by the computer of response message packets responding to the address inquiry message packets.

10 In embodiments of the invention, the iteration control module is configured to enable the
11 computer to provide at least a minimum time period for each iteration, and to extend the time period
12 if the computer receives response message packets during the iteration. In addition, the iteration
13 control module is configured to adjust the minimum time period in relation to the time delay
14 between the time the computer transmits address inquiry message packets and the time it receives
15 respective address inquiry message packets responsive thereto.

16 In further embodiments of the invention, the computer discovers predetermined
17 characteristics of devices which provide response message packets, including whether they are
18 running a port mapper.

19 BRIEF DESCRIPTION OF THE DRAWINGS

20 This invention is pointed out with particularity in the appended claims. The above and
21 further advantages of this invention may be better understood by referring to the following
22 description taken in conjunction with the accompanying drawings, in which:

23 FIG. 1 is a schematic diagram of a computer network including a computer constructed in
24 accordance with the invention;

1 FIG. 2 is a functional block diagram of a computer useful in the network depicted in FIG.
2 1 including an arrangement for facilitating efficient determination by a computer connected in a
3 computer network of network addresses used by other computers and other devices connected in the
4 network, and of determining predetermined characteristics of those other computers and other
5 devices, constructed in accordance with the invention; and

6 FIGS. 3A through 3F together comprise a flowchart useful in understanding operations
7 performed by the computer depicted in FIG. 2 in connection with determining network addresses
8 of the other computers and other devices connected in the network depicted in FIG. 1.

9 DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

10 FIG. 1 depicts a computer network 10 including a computer which, in turn, includes an
11 arrangement for facilitating efficient determination by a computer connected in a computer network
12 of network addresses used by other computers and other devices connected in the network, and of
13 determining predetermined characteristics of those other computers and other devices, constructed
14 in accordance with the invention. With reference to FIG. 1, network 10 includes a plurality of
15 computers 11(1) through 11(N) (generally identified by reference numeral 11(n)) and 12 which are
16 interconnected by a communication link 13. As is conventional, the computers 11(n) and 12 are of
17 the conventional stored-program computer architecture. At least some of the computers 11(n) are
18 in the form of personal computers or computer workstations, each of which includes a system unit,
19 a video display unit and operator input devices such as a keyboard and mouse. The computer 12 also
20 includes a system unit, and may also include a video display unit and operator input devices. A
21 system unit generally includes processing, memory, mass storage devices such as disk and/or tape
22 storage elements and other elements (not separately shown), including network interface elements
23 for interfacing the respective computer system 11(n) or server computer 12 to the communication
24 link 13. A video display unit permits the computer to display processed data and processing status
25 to the user, and an operator input device enable the user to input data and control processing by the

-7-

1 computer. The computers 11(n) and 12 transfer information, in the form of messages, through their
2 respective network interface devices among each other over the communication link 13. The
3 communication link 13 interconnecting the client computers 11(n) and server computer 12 in the
4 network 10 may, as is conventional, comprise wires, optical fibers or other media for carrying
5 signals representing information among the computers 11(n) and 12.

6 In one embodiment, the network 10 is organized in a "client-server" configuration, in which
7 at least one computer, namely, computer 12, operates as a server, and the other computers 11(n)
8 operate as clients. Typically, the servers include large-capacity mass storage devices which can store
9 copies of programs and data which are available for retrieval by the client computers over the
10 communication link 13 for use in their processing operations. From time to time, a client computer
11 system 11(n) may also store data on the server computer 12, which may be later retrieved by it (the
12 client computer that stored the data) or other client computers for use in their processing operations.
13 The server computers may be generally similar to the client computers 11(n), including a system
14 unit, video display unit and operator input devices and may be usable by an operator for data
15 processing operations in a manner similar to a client computer. Alternatively, at least some of the
16 server computers may include only processing, memory, mass storage and network interface
17 elements for receiving and processing retrieval or storage requests from the client computers, and
18 generating responses thereto.

19 The network 10 may be connected to a plurality of other networks (not shown), each of
20 which may be structured and organized similarly to network 10, to form a unitary "wide area"
21 network. Each network, including network 10, may be divided into a hierarchy comprising a
22 plurality of sections, sub-sections and so forth, which may be illustratively termed "sub-nets," "sub-
23 sub-nets," and so forth, in one or more levels. Each computer system 11(n) and server computer 12
24 is identified by a network address which is structured to reflect the network division hierarchy. For
25 example, each computer system's network address includes a series of one or more high-order
26 portions, which can identify the particular network, sub-net, sub-sub-net, and so forth, in the
27 hierarchy, and a low-order portion which provides an identification for the particular computer

system 11(n) or server computer 12 in the lowest-level sub-net. Thus, each computer system's network address may have a structure $\langle a_{B_N}^N \cdots a_0^N | a_{B_{N-1}}^{N-1} \cdots a_0^{N-1} | \cdots | a_{B_0}^0 \cdots a_0^0 \rangle$, where " $a_{B_N}^N \cdots a_0^N$ " corresponds to the series of " B_N+1 " address bits identifying, for example, the network within the wide-area network, " $a_{B_{N-1}}^{N-1} \cdots a_0^{N-1}$ " corresponds to the series of " $B_{N-1}+1$ " address bits identifying a second-level sub-net (corresponding to the highest sub-net level in the network hierarchy), and so forth, and the vertical bar "|" indicates that the successive series of address bits are concatenated into a single address. All of the possible binary-encoded values represented by the entire address, that is $\langle a_{B_N}^N \cdots a_0^0 \rangle$, define a network address space, and a computer 11(n) can have any network address in that network address space. Thus, if there are "M" bits in the network address, there will be 2^M possible addresses in the address space.

Different networks in a wide-area network may have different numbers of sub-net levels, and different numbers of sets of the bits $\langle a_{B_{N-1}}^{N-1} \cdots a_0^0 \rangle$ below the network-identification level which are allocated to sub-net identification. Indeed, networks in the wide-area network need not have any sub-net levels. In these cases, the computers will have "M" address bits, with the high-order B_N+1 bits identifying the network in the wide-area network. However, the other address bits $\langle a_{B_{N-1}}^{N-1} \cdots a_0^0 \rangle$ may be allocated to sub-net or other levels (if any) in a manner which may differ from other networks in the wide area network.

Each computer connected to the communication link 13 can transfer information to another computer connected to the communication link by transmitting message packets over the communication link 13. Each message packet includes an address portion which contains the address of the network, sub-net, sub-sub-net, and so forth, as well as the address for the specific computer system 11(n) or server computer 12 which is to receive the message packet. Generally, computers 11(n) which are associated with the same sub-net are typically located proximate one

1 another, such as in the same building or portion of a building. On the other hand, sub-nets associated
2 with the same network may be located in widely-divergent locations.

3 The invention, in one aspect, provides an efficient network address discovery mechanism by
4 which a computer connected to the network can determine which of the network addresses in the
5 network's address space is used by computers or other resources and devices (generally,
6 "computers") that are connected to the network. In another aspect, the invention provides a
7 mechanism by which the computer, after determining that an address is used by a computer,
8 determines predetermined characteristics of the computer, in particular the types of methods or
9 procedures which are available from the respective computer by means of remote method
10 invocations or RPC (remote procedure calls).

11 In connection with the first aspect, that is, the determination of the addresses in the address
12 space which are used by other computers, the computer that is attempting to determine those
13 addresses, illustratively, computer 11(n'), transmits a plurality of address inquiry message packets
14 over the communication link 13, each message with one of the possible address from the address
15 space. In one embodiment, in which message packets are transferred over the network using the
16 Internet protocol ("IP"), the address inquiry message packets may be in the form of ICMP (Internet
17 Control Message Packet) echo request, or "ping" packets [see, for example, R. Perlman,
18 Interconnections: Bridges and Routers, (Addison-Wesley, 1992) pp. 185-189]. If the address in the
19 address inquiry message packet corresponds to the network address of another computer, such as
20 computer 11(n") (n"≠n') in the network, the other computer 11(n") will generate a response message
21 packet and transmit it back to the computer 11(n'). In the embodiment, when the computer 11(n')
22 receives the response message packet, it will record in, for example, a database, the fact that the
23 address is used by another computer in the network. If the computer 11(n') does not receive a
24 response message packet, the computer 11(n') can, but need not, record the fact that no response
25 message packet was received. The fact that the computer 11(n') did not receive a response message
26 packet may indicate, for example, that no computer is connected to the network which uses the
27 address. However, the fact that the computer 11(n') did not receive a response message packet may

-10-

1 also indicate that a computer 11(n'') connected to the network uses the address but that the computer
2 11(n'') did not receive the address inquiry message packet (and thus would not generate a response
3 message packet) because it (that is, the address inquiry message packet) was lost, or that the
4 computer 11(n'') did receive the address inquiry message packet and transmitted a response message
5 packet, but that the computer 11(n') that transmitted the address inquiry message packet did not
6 receive the address inquiry message packet because it (that is, the address inquiry message packet)
7 was lost. To accommodate such possible packet loss, the computer 11(n') may, if it does not receive
8 a response message packet relating to a particular network address within a predetermined time
9 interval after it transmits an address inquiry message packet, at some point later transmit another
10 address inquiry message packet using the same network address. The computer 11(n') may perform
11 these operations several times. If the computer 11(n') receives a response message packet for the
12 network address in response to any of the address inquiry message packets that it transmits, it (that
13 is, the computer 11(n')) will record in its database the fact that the address is used by another
14 computer 11(n'') in the network 10.

15 The computer 11(n'), when it is transmitting address inquiry message packets to determine
16 the network addresses for which other computers are respective destinations, neither

17 (i) waits, after performing all of the operations described above in connection with one
18 network address before transmitting address inquiry message packets for a next network address in
19 the address space, since that could require an unduly long period of time to accommodate all of the
20 network addresses in the address space, nor

21 (i) transmits address inquiry message packets for all possible network address in the address
22 space, since all of the other computers in the network will receive them and transmit response
23 message packets substantially simultaneously, resulting in congestion and likely lost message
24 packets at the computer system 11(n).

25 Instead, the computer 11(n') transmits address inquiry message packets in a series of iterations. In
26 each iteration, the computer 11(n') transmits a set of address inquiry message packets, each of which

-11-

1 is associated with a network address. The computer 11(n') will wait for a time during which it can
2 receive response message packets that are transmitted in response to the address inquiry message
3 packets that were transmitted during the iteration, as well as response message packets that were
4 transmitted in response to any address inquiry message packets that were transmitted during previous
5 iterations but not received during those iterations. In each iteration, the network addresses that are
6 used in the address inquiry message packets are preferably sequential, and may be associated with
7 computers associated with the same sub-net, although, depending on the number of address inquiry
8 message packets transmitted during an iteration, several iterations may be required to transmit
9 address inquiry message packets for all network addresses in a sub-net. In successive iterations, the
10 address inquiry message packets will preferably be from successive sequences of network addresses
11 in the network address space.

12 In accordance with the invention, the amount of time that the computer 11(n') waits after
13 transmitting a set of address inquiry message packets during an iteration before it begins transmitting
14 address inquiry message packets for the next iteration may vary. The amount of time that the
15 computer 11(n') will wait before it begins transmitting the next set of address inquiry message
16 packets depends on whether it receives, during an initial minimum time period, any response
17 message packets for the set of message packets that it transmitted during the iteration.

18 If, during an iteration, the computer 11(n') does receive at least one such response message
19 packet for the address inquiry message packets transmitted at the beginning of the iteration, but not
20 response message packets for all of the address inquiry message packets, it can wait for an additional
21 incremental time period before beginning the next iteration. If, during the additional incremental
22 time period, the computer 11(n') again receives at least one response message packet for the set of
23 address inquiry message packets transmitted at the beginning of the iteration, and if it has not
24 received response message packets for all such address inquiry message packets, it can wait for yet
25 another incremental time period. These operations can be repeated through additional incremental
26 time periods, up to a predetermined maximum time period. On the other hand, if, during the initial
27 minimum time period, the computer 11(n') does not receive any response message packets for the

-12-

1 address inquiry message packets that it transmitted at the beginning of the iteration, it can begin the
2 next iteration and transmit the next set of address inquiry message packets. Similarly, if, during any
3 incremental time period after the minimum time period, the computer 11(n') does not receive a
4 response message packet for the address inquiry message packets that it transmitted at the beginning
5 of the iteration, it can at that point begin the next iteration and transmit the next set of address
6 inquiry message packets. In any case, the computer 11(n') will begin the next iteration no later than
7 the maximum time period. Thus, the amount of time that the computer 11(n') will wait, after it
8 transmits the address inquiry message packets for one iteration, before beginning transmission of
9 address inquiry message packets for the next iteration, can vary from a minimum time period to a
10 maximum time period, in increments defined by the incremental time period, depending on whether
11 the computer 11(n') receives, during the minimum time period or each successive incremental time
12 period, a response message packet responding to an address inquiry message packet transmitted at
13 the beginning of the iteration.

14 For any response message packets that the computer 11(n') receives during the iteration,
15 including response message packets which were in response to address inquiry message packets that
16 transmitted by the computer 11(n') during a previous iteration, it (that is, the computer 11(n')) will
17 receive them (that is, the response message packets) and record the network addresses in the
18 response message packets as being used by computers 11(n''), (11n'''), ..., (n'', n''' ≠ n') on the network.

19 These operations continue until the computer 11(n') has transmitted address inquiry message
20 packets for all of the network addresses in the network address space. For network addresses for
21 which the computer 11(n') did not receive response messages packets in response to address inquiry
22 message packets, the computer 11(n') can perform a retry operation during which it repeat the above-
23 described operations in connection with those network addresses. It will be appreciated that the
24 computer 11(n') may, during the retry operation, receive response message packets in response to
25 address inquiry message packets which it had previously transmitted during the original operation;
26 in that case, the computer 11(n') can record the network address associated with the response
27 message packet as being used in the same manner as it would with response message packets in

-13-

1 response to address inquiry message packets which were transmitted during the retry operation. The
2 computer 11(n') can repeat these operation through a predetermined maximum number of retry
3 operations, performing each retry operation in connection with network addresses for which it has
4 not previously received response message packets. For those network addresses for which the
5 computer

6 A benefit of adjusting the length of the time period is as follows. Typically, the address
7 space of a network is relatively sparsely populated, that is, there are relatively few network addresses
8 which are actually being used by computers and other devices which are connected to the network.
9 However, those network addresses which are used are generally clustered and within a cluster the
10 network addresses which are used are relatively thickly populated. For unpopulated regions of the
11 network address space, during each iteration the computer 11(n') will only wait for the initial
12 minimum time period before beginning the next iteration and so, if the initial minimum time period
13 is a relatively small portion of the maximum time period, the iterations will be relatively short and
14 the computer 11(n') to sequence through those unpopulated regions relatively quickly. However,
15 when the computer 11(n'), during an iteration, is operating in a relatively thickly populated region
16 of the network's address space, by increasing the time period incrementally while the computer
17 11(n') is receiving response message packets, before sending out the next set of address inquiry
18 message packets for the next iteration, the computer 11(n') will have more time to receive and
19 process the response message packets which it receives during the iteration, effectively adapting the
20 time period (either the initial minimum time period or the initial period as incremented) which the
21 computer 11(n') will wait before sending the next set of address inquiry messages for the next
22 iteration. This can reduce the congestion at the computer 11(n') and also reduce the likelihood of
23 lost response message packets. In addition, making sure that the computer 11(n') does not wait
24 beyond the maximum time period before transmitting address inquiry message packets for the next
25 iteration ensures that it (that is, the computer 11(n')) does not wait too long before starting the next
26 iteration, which can also unduly extend the network address discovery operation.

As a refinement, the initial minimum time period used by the computer 11(n') can also vary. In that connection, when for each sub-net, the computer 11(n') maintains a running average for the round-trip time for message packets, from the time the address inquiry message packets are transmitted to the time any response message packets are received. If the running average round-trip time exceeds either the minimum time period, or the time period as incremented as described above, for the next iteration the computer 11(n') will set as the minimum time period the current minimum time period incremented by the incremental time period, provided the new minimum time period does not exceed the maximum time period. The computer 11(n') will use that incremented initial minimum time period as the initial minimum time period during the next iteration. As noted above, within a populated cluster of the network's address space, the network addresses are relatively thickly populated. Typically, each such cluster will be associated with a sub-net, sub-sub-net, and so forth (generally, "sub-net"). Since a sub-net may be relatively close to, or distant from, the sub-net to which includes computer 11(n'), the propagation delay for address inquiry message packets and respective response message packets may be correspondingly short or long. After the computer 11(n') has completed one or more iterations during which it transmits address inquiry message packets for network addresses in a cluster associated with a sub-net, it will reset the initial minimum time interval to the predetermined initial minimum time interval. If the computer 11(n'), during an iteration in connection with network addresses which are associated with a sub-net, determines that it has received response message packets for which the running average round trip time exceeds the current minimum time period, it is likely that the sub-net for which it is currently transmitting address inquiry message packets is relatively distant, with a correspondingly long propagation delay. In that case, adjusting the initial minimum time delay as described above will serve to adapt the initial minimum time interval to accommodate the increased propagation delay on a sub-net by sub-net basis. Resetting the initial minimum time interval after finishing iterations in connection with a sub-net will ensure that the initial minimum time interval does not remain at the incremented length for iterations during which address inquiry message packets are transmitted using network addresses in unpopulated regions of the network address space or using network addresses for sub-nets for which the propagation delay is relatively short.

1 In another aspect of the invention, the computer 11(n'), after determining at least one network
2 address in the network address space which used by another computer 11(n'') (n'' ≠ n'), can efficiently
3 determine predetermined characteristics of the other computer. In that connection, the computer
4 11(n') determines whether the other computer 11(n'') is running a port mapper, which is useful in
5 connection with requesting processing services from the other computer 11(n'') by way of a remote
6 procedure call ("RPC"). The computer 11(n') determines whether the other computer 11(n'') is
7 running a port mapper by transmitting a message packet therefor, and so it can perform these
8 operations at any time after it determines that another computer 11(n'') is using a particular network
9 address as described above. In one embodiment, the computer 11(n') can perform operations in
10 connection with the second aspect on an interleaved basis with the operations described above in
11 connection with the first aspect.

12 FIG. 2 depicts a functional block diagram of components of the computer 11(n') which are
13 related to the invention as described above. With reference to FIG. 2, the computer 11(n') includes
14 a network discover system 20, an operating system 21 and an network interface 22. The computer
15 11(n') can also include other components (not shown) which are conventional in a computer,
16 including components for performing processing operations, storing information, receiving
17 information from an operator or displaying information to an operator and the like. The network
18 interface 22 transmits message packets over, and receives message packets from, the communication
19 link 13 (FIG. 1). The network interface 22 can receive message packets, including address inquiry
20 message packets, to be transmitted from the operating system 21, and provide message packets
21 received from the communication link 13 to the operating system 21. The operating system 21, in
22 turn, will generally receive information to be transferred in message packets from other components
23 in the computer 11(n'), including the network discover system 20. After the operating system 21
24 receives the information to be transferred it generates the message packets, which it provides to the
25 network interface 22 for transfer over the communication link 13. For message packets received
26 from the network interface 22, the operating system 21 will determine the appropriate component,

-16-

1 such as network discover system 20, which is to receive the information, and provide the information
2 thereto.

3 The network discover system 20 includes an occupied address discover module 23, a port
4 mapper discover module 24, and a discover database 26, all of which operate under control of a
5 control module 27. The discover database 26 stores information concerning aspects of the network
6 which have been determined by the network discover module system 20, including, in connection
7 with the invention, the network addresses of the network address space which are occupied, that is,
8 for which there are computers which are identified thereby, and whether each such computer is
9 running a port mapper. The occupied address discover module 23 controls transmission of address
10 inquiry message packets, and receives information related to response message packets and loads
11 occupied network address information into the discover database 26. The port mapper discover
12 module 24 uses the occupied network address information from the discover database 26 to control
13 generation of port mapper status request message packets for the occupied network addresses, and
14 receives information responsive thereto, which it stores in the discover database 26.

15 As noted above, the control module 27 control the occupied address discover module 23, and
16 port mapper discover module 24. In particular, the control module 27 will receive, either from an
17 operator through an user interface, or from another component, such as a program (not shown) being
18 executed by the computer 11(n'), identification of the portion or portions of the network address
19 space for which address inquiry message packets are to be generated, and enable the occupied
20 address discover module 23 to operate in connection therewith. In one embodiment, the control
21 module 27 may provide the information as to the portion or portions of the network address space
22 to the occupied address discover module 23 in the form of address space ranges or in the form of a
23 table of network addresses, or both. Details of operations performed by the occupied address
24 discover module 23 in generating address inquiry message packets, receiving response message
25 packets and storing occupied network address information in the discover database, will be described
26 below in connection with FIGS. 3A through 3F.

-17-

1 The port mapper discover module 24 determines whether computers, which occupy network
2 addresses discovered by the occupied address discover module 23, are running respective port
3 mappers. In that operation, the port mapper discover module 24 enables, for respective occupied
4 network addresses, the generation of RPC (remote procedure call) null message packets, which are
5 transmitted by the network interface 22 over the communication link 13. For each RPC null
6 message packet, the computer 11(n") that receives the packet will respond and condition a flag in
7 the response indicating whether it (that is, the computer 11(n")) is running a port mapper. The port
8 mapper discover module 24 can store that information in the discover database 26.

9 As noted above, operations performed by the occupied address discover module 23 in
10 determining the occupied network addresses in the network address space will be described in
11 connection with the flowchart depicted on FIGS. 3A through 3F. With reference to FIG. 3F, after
12 the occupied address discover module 23 receives from the control module 23 either a range of the
13 network address space (which may include the entire network address space) or a table of network
14 addresses which are to be "pinged" to determine which network addresses are used by computers
15 connected to the network, a retry number, and the minimum, incremental and maximum time periods
16 (step 100), it (that is, the occupied address discover module 23) begins one or more retry operations
17 as determined by the retry number received in step 100 to determine which network addresses are
18 occupied.

19 In each retry operation, the occupied address discover module 23 performs one or more
20 iterations, with the number of iterations being determined by the number of network addresses in the
21 range or table and the number of network addresses for which address inquiry message packets are
22 to be generated in each iteration. In each iteration, the occupied address discover module 23 initially
23 selects a series of a predetermined number of network addresses from the range/table for which
24 response message packets have not yet been received (step 101) and generates for each of the
25 selected network addresses an address inquiry message packet for transfer by the network interface
26 22 over the communication link 13 (step 102). Each address inquiry message packet includes one
27 of the selected network addresses and a time stamp; the time stamp will be used in determining the

1 running average round-trip time for the use in adjusting the initial minimum time period described
2 above.

3 After the address inquiry message packets have been transferred, the occupied address
4 discover module 23 determines whether the network addresses for which the address inquiry
5 message packets were generated in step 102 is for the last set of network addresses in the range/table
6 for the retry operation (step 103). If the occupied address discover module 23 makes a negative
7 determination in step 103, it will load the initial minimum time period value into a termination time
8 interval register (step 104). On the other hand, if the occupied address discover module 23 makes
9 a positive determination in step 103, it will load the maximum time period value into the termination
10 time interval register (step 105). The time period value stored in the termination time interval
11 register will be used in determining the time period for the iteration, and by loading the maximum
12 time period value into the termination time value register for the last iteration of the retry operation
13 (reference step 105), that will ensure that the maximum time will be allowed to receive response
14 message packets during retry operation.

15 After the address inquiry message packets have been transferred, the occupied address
16 discover module 23 initializes two timers, namely, an iteration time interval timer and a time interval
17 extension timer, and enables them to start operating (step 106). The iteration time interval timer
18 will be used in determining when the iteration is to end. The time interval extension timer will be
19 used in determining when the initial minimum time period or each incremental time period has
20 ended; this will be used in controlling extension of the iteration time period to ensure that the
21 iteration time period is extended only once during the initial minimum time period or each
22 incremental time period even if multiple response message packets are received during the initial
23 minimum time period or respective incremental time period. The iteration time interval timer may,
24 for example, be initialized at zero and enabled to increment, and the time interval extension timer
25 may be initialized at the initial minimum time period and be enabled to decrement.

1 After the occupied address discover module 23 has started the iteration time interval timer
2 and the time interval extension timer in step 104, if it (that is, the occupied address discover module
3 23) determines that a response message packet has been received by the network interface from the
4 communication link 13, and addressed to the computer 11(n') (step 107), it (that is, occupied address
5 discover module 23) will enable an indication that the network address in the response message
6 packet is occupied to be stored in the discover database 26 (step 108). After a network address has
7 been indicated as being occupied, the other modules of the network discover system 20, including
8 the port mapper discover module 24 can perform operations therewith as described above.
9 Following step 108, the occupied address discover module 23 can determine whether the response
10 message packet was transmitted in response to an address inquiry message packet that was
11 transferred in step 102 of the current iteration (step 109). If the occupied address discover module
12 23 makes a positive determination in step 109, it will determine the condition of an increment time
13 interval disable flag (step 110), which will control incrementing the iteration time interval for the
14 iteration.

15 If the occupied address discover module 23 determines in step 110 that the increment time
16 interval disable flag is set, which will occur if the response message packet received in step 107 is
17 the first response message packet that the occupied address discover module 23 received during the
18 respective initial minimum time interval or current incremented time interval, that was transmitted
19 in response to an address inquiry message packet that had been transferred in step 102 of the current
20 iteration, the occupied address discover module 23 will increment the time interval value stored in
21 the termination time interval register (step 111) and set the increment time interval disable flag (step
22 112). In addition, the occupied address discover module 23 will compare the time interval value
23 stored in the termination time interval register to the maximum time interval (step 113) and, if the
24 time interval value stored in the termination time interval register is greater than the maximum time
25 interval, copy the maximum time interval into the termination time interval register (step 114). The
26 increment time interval disable flag is provided to ensure that the time interval value stored in the
27 termination time interval register is incremented only once for response message packets received

1 during the initial minimum time period or respective incremented time period for the iteration. Step
2 114 is provided to ensure that the time interval for the iteration is limited to the maximum time
3 interval.

4 If the occupied address discover module 23 determines in step 110 that the increment time
5 interval disable flag is clear, which will occur if the response message packet received in step 107
6 is not the first response message packet that the occupied address discover module 23 received
7 during the respective initial minimum time interval or current incremented time interval, that was
8 transmitted in response to an address inquiry message packet that had been transferred in step 102
9 of the current iteration, it will skip steps 111 through 114, to avoid incrementing the time interval
10 value stored in the termination time interval register.

11 Returning to step 109, if the occupied address discover module 23 makes a negative
12 determination in that step, indicating that a response message packet had been received, but that the
13 response message packet was not in response to an address inquiry message packet that was
14 transferred in step 102 of the current iteration, or step 114, the occupied address discover module
15 23 will determine whether the response message packet was for the same sub-net as the sub-net
16 associated with the address inquiry message packets which were transferred in step 102 (step 115).
17 If the occupied address discover module 23 makes a positive determination in step 115, it will update
18 a running average round-trip time based on the time stamp in the response message packet (which
19 is a copy of the time stamp in the address inquiry message packet for which the response message
20 packet is a response) and the time at which the response message packet was received by the
21 computer 11(n') (step 116).

22 Following

23 (i) step 116;

-21-

1 (ii) step 115, if the occupied address discover module 23 makes a negative determination in
2 that step, indicating that the response message packet was for the same sub-net as the sub-net
3 associated with the address inquiry message packets which were transferred in step 102;

4 (iii) step 113, if the occupied address discover module 23 incremented the time interval value
5 in the termination time interval register in that step;

6 (iv) step 110, if the occupied address discover module 23 determined in that step that the
7 increment time interval disable flag is set, indicating that the time interval value in the termination
8 time interval register is not to be incremented, or

9 (v) step 107 if the occupied address discover module 23 makes a negative determination in
10 that step, indicating that a response message packet has not been received following activation of
11 the counter,

12 the occupied address discover module 23 will determine whether the time interval value provided
13 by the iteration time interval timer corresponds to the time interval value stored in the termination
14 time interval register (step 117). If the occupied address discover module 23 makes a negative
15 determination, that is, if it determines that the time interval value provided by the timer is less than
16 the time interval value provided by the termination time interval register, it (that is, the occupied
17 address discover module 23) will determine whether the time interval extension timer has timed out
18 (step 118). If the occupied address discover module 23 makes a positive determination in step 118,
19 that is, if it determines that the time interval extension timer has timed out, it will reset the increment
20 time interval disable flag (step 119) and re-initialize the time interval extension timer with the
21 increment time period value and enable it to start decrementing (step 120). By resetting the
22 increment time interval disable flag in step 119, the occupied address discover module 23 may again
23 increment the time period value for the iteration in the termination time period register in step 111
24 if another response message packet is received that was transmitted during step 102 of the current
25 iteration.

-22-

1 Following

2 (i) step 120, or

3 (ii) step 118, if a negative determination is made in that step, indicating that the respective
4 initial minimum time period or incremental time period has not passed,
5 the occupied address discover module 23 will return to step 107 to wait for another response
6 message packet.

7 The occupied address discover module 23 will iteratively perform steps 107 through 120
8 until it determines, in step 117, that the time interval value provided by the timer corresponds to the
9 time interval value stored in the termination time interval register. When that occurs, the iteration
10 terminates and the occupied address discover module 23 prepares to perform another iteration in
11 connection with a next set of network addresses, if any, in the network address range or table. In
12 those operations, the occupied address discover module 23 will initially determine whether there are
13 any additional network addresses in the address range or table (step 121). If the occupied address
14 discover module 23 makes a positive determination in step 121, which will occur if there are
15 additional network addresses in the address range or table, it will determine whether the set of
16 network addresses to be used in the next iteration are for the same sub-net as for the network
17 addresses used for the current iteration (step 122). As indicated above, the network addresses to be
18 used in the next iteration will be for the same sub-net if predetermined ones of the bits of the network
19 address represent the same value. If the occupied address discover module 23 determines in that step
20 that the set of network addresses to be used in the next iteration are not for the same sub-net as for
21 the network addresses used for the current iteration, it will reset the initial minimum time period to
22 its original value (step 123) and return to step 101 to begin the next iteration.

23 In addition, returning to step 121, if the occupied address discover module 23 makes a
24 negative determination in that step, all of the network addresses have been used in address inquiry
25 message packets for the retry operation. If the occupied address discover module 23 makes a

1 negative determination in step 121, it will determine whether to perform another retry operation.
2 Initially, the occupied address discover module 23 will decrement the retry number received in step
3 100 (step 124), and determine whether the decremented retry number corresponds to a predetermined
4 retry termination criterion (step 125). If the occupied address discover module 23 makes a negative
5 determination in step 125, that is, if it determines that the decremented retry number does not
6 correspond to the retry termination criterion, it will then determine whether it has received response
7 message packets in connection with all of the network addresses in the range/table that it received
8 in step 100 (step 126). If the occupied address discover module 23 makes a negative determination
9 in step 126, it will return to step 101 to perform another retry operation.

10 The occupied address discover module 23 will perform steps 101 through 126 through one
11 or more retry operations, until it makes either

12 (i) a positive determination in step 126, indicating that it had received response message
13 packets for all of the network addresses in the range or table that it received in step 100, or

14 (ii) a determination in step 125 that the decremented retry number corresponds to the
15 predetermined retry termination criterion.

16 In either case (i) or (ii), the occupied address discover module 23 will terminate operations (step
17 127).

18 It will be appreciated that numerous modifications may be made to the network discover
19 arrangement and system as described above in connection with FIGS. 1 through 3F. For example,
20 although the network discover system has been described in connection with a client computer
21 11(n'), it will be appreciated that such a system may also be used in connection with a server
22 computer such as computer 12, or a computer which communicates with other computers on a peer-
23 to-peer basis. In addition, such a system may also be used in connection with non-computer
24 resources and devices which are connected in the network.

1 Further, although the network discovery system has been described as comprising
2 components 23 through 25 for performing respective operations as described above, it will be
3 appreciated that a network discovery system in connection with the invention may have a subset of
4 these components or additional components. In addition, although the occupied address discover
5 module 23 has been described as providing adaptable-length time intervals for respective iterations,
6 to facilitate lengthening of the time period of an iteration both when the computer 11(n') is
7 transferring address inquiry message packets for network addresses in a populated region of the
8 network address space (reference steps 107 through 112 and 116 through 119), and for network
9 addresses in a populated region for which the sub-net is located relatively distant from the computer
10 11(n'), with correspondingly-increased propagation delays (reference steps 113 through 115 and 121
11 through 124), it will be appreciated that the occupied address discover module 23 may only lengthen
12 the time period for one of these.

13 It will be appreciated that a system in accordance with the invention can be constructed in
14 whole or in part from special purpose hardware or a general purpose computer system, or any
15 combination thereof, any portion of which may be controlled by a suitable program. Any program
16 may in whole or in part comprise part of or be stored on the system in a conventional manner, or it
17 may in whole or in part be provided in to the system over a network or other mechanism for
18 transferring information in a conventional manner. In addition, it will be appreciated that the system
19 may be operated and/or otherwise controlled by means of information provided by an operator using
20 operator input elements (not shown) which may be connected directly to the system or which may
21 transfer the information to the system over a network or other mechanism for transferring
22 information in a conventional manner.

23 The foregoing description has been limited to a specific embodiment of this invention. It will
24 be apparent, however, that various variations and modifications may be made to the invention, with
25 the attainment of some or all of the advantages of the invention. It is the object of the appended
26 claims to cover these and such other variations and modifications as come within the true spirit and
27 scope of the invention.

1

What is claimed as new and desired to be secured by Letters Patent of the United States is:

2000-01-01 10:00:00